

## **User Interface for Controlling Access to Computer Objects**

### **Technical Field**

**[0001]** The present invention relates to user interfaces for accessing and controlling access to computer objects and, in particular, to a user interface for accessing or controlling access to computer objects in computer spaces relating to computer locations and computer users.

### **Background and Summary**

**[0002]** Computer objects (e.g., files) are commonly copied or saved from one computer to another. Computer messages are sent between computers. Computer users give each other access to or “share” different sets of computer files or different computer file directories.

**[0003]** Typically, the sharing of computer objects between users as a “share” is distinct from the saving of computer objects to computer (e.g. network) locations. This distinction relates to the traditional view that computer storage occurs at locations (e.g., drives, folders, etc.), while the sharing of computer objects relates to the security or accessibility of the object at that location. As a result, the user interfaces for saving computer objects to computer locations are distinct from the user interfaces for setting the sharing of the objects.

**[0004]** An aspect of the present invention is an appreciation that access to a computer object according to its computer location is analogous to who is accessing the object. The common aspect is accessibility to the object. Accordingly, the present invention provides a computer object access control graphical user interface for setting computer locations where a computer object may be accessed and users who may access the computer object. The computer object may be a computer file, a computer message, person contact information, etc.

**[0005]** The access control user interface may operate, for example, to save a computer object to one or more computer locations, and also to specify sharing of the computer object with one or more other users so that they can access the object. Saving a computer object and setting the sharing of it are together referred to as controlling access to the object, the former referring to where the object may be accessed from and the latter referring to who may access the object.

**[0006]** In one implementation, the user interface includes a name field indicating a name for the computer object and multiple access control fields that are rendered together and indicate multiple selectable computer spaces for the computer object. At least one of the computer spaces corresponds to a computer location, and at least one of the computer spaces corresponds to access to the computer object for one or more computer users. The present invention provides, therefore, a unified interface for selecting and displaying destinations that can be either a file store or a person, thereby unifying the control of access to a computer object.

**[0007]** In one implementation, the multiple computer spaces are listed in a ranked sequence, which can be based upon recency of user interaction with the computer spaces (i.e., locations or users) or according to associations to the computer object determined automatically from user computer interactions. Users can be represented by an email address, an instant messaging name, a full name, or any other person identifiers, and specific transports to the users can also be identified or specified.

**[0008]** Additional objects and advantages of the present invention will be apparent from the detailed description of the preferred embodiment thereof, which proceeds with reference to the accompanying drawings.

### **Brief Description of the Drawings**

**[0009]** Fig. 1 is a an illustration of a computer object access control graphical user interface for setting computer locations where a computer

object may be accessed and users who may access the computer object.

[0010] Fig. 2 is a an illustration of another computer object access control graphical user interface for setting computer locations where a computer object may be accessed and users who may access the computer object.

[0011] Fig. 3 is a an illustration of a computer object access control graphical user interface for accessing a computer object computer from a location on a computer or received via computer communication.

[0012] Fig. 4 is a flow diagram of a computer-implemented computer object access control graphical user interface method.

[0013] Fig. 5 is a functional block diagram of an automatic context association system that automatically determines context associations between user computer activities, computer objects (e.g., files, contacts, etc.), and other events.

[0014] Fig. 6 is a detailed block diagram of one implementation of data collection system.

#### **Detailed Description of Preferred Embodiments**

[0015] Fig. 1 is a an illustration of a computer object access control graphical user interface 100 for setting computer locations where a computer object may be accessed and users who may access the computer object. The computer object may be a computer file, a computer message, person contact information, etc.

[0016] Access control user interface 100 may operate, for example, to save a computer object to one or more computer locations, and also to specify sharing of the computer object with one or more other users so that they can access the object. Saving a computer object and setting the sharing of it are together referred to as controlling access to the object, the former referring to where the object may be accessed from and the latter referring to who may access the object. Access control user interface 100 may be generated from within any application or by a

computer operating system whenever the access controlling user wishes to save the object to a specified location of share access with a specified other person.

**[0017]** Access control user interface 100 includes a name field 102 in which a name for the computer object can be entered, either automatically by a computer or manually by the user. An author field 104 specifies who created the computer object (e.g., a computer file, a computer message, person contact information, etc.) for which access is being set with user interface 100. An optional comment field 106 allows the user who is controlling access to the object (referred to as the access-controlling user) to enter and store a comment about the object, and an optional “more” link 108 gives the user access to optional additional fields for entering other properties relating to the object. The additional fields at “more” link 108 may include any or all of a creation date field indicating when the object was created, a modified field indicating when the object was last modified, any keywords, category names, etc.

**[0018]** An access control field 110 includes a listing or display of multiple computer spaces 112A-112F (referred to generally as computer spaces 112) that have or can be selected to have access to the computer object. Computer spaces 112 may refer to different portions of a computer network (e.g., public and private portions or “folders”), different computers, different computer access rights of different computer users, different computer users, etc. User interface 100 provides, therefore, a unified control over access to the computer object at computer spaces corresponding to computer users and computer locations.

**[0019]** For example, computer spaces 112A, 112B, and 112F reference computers or other locations on a computer network where access to the object may be provided. Computer space 112A indicates the controlling user’s local or desktop computer, computer space 112B indicates the controlling user’s portable or laptop computer, and computer space 112F

indicates a location on a local or global computer network. It will be appreciated that these particular computer spaces 112A, 112B, and 112F are merely examples.

**[0020]** As additional examples, computer spaces 112C-112E reference computer users to whom access to the object may be granted. Computer space 112C indicates an individual (e.g., Coworker1) to whom the object is provided by email. Computer space 112D indicates a group of users (e.g., "Work Group") to whom the object may be provided, either by computer communication such as email or at a shared computer location. Computer space 112D indicates a group of instant messaging users or "buddies" who could be given access. The users of computer spaces 112C-112E can each be represented by an email address, an instant messaging name, a full name, or any other person identifiers, and specific transports to the users can also be identified or specified. A scroll bar 113 allows the access controlling user to scroll to other selectable computer spaces.

**[0021]** Access control indicators 114A-114F indicate whether each corresponding computer space 112A-112F has been selected to have access to the computer object. For example, access control indicators 114A-114C indicate that respective computer spaces 112A-112C have access to the computer object, and access control indicators 114D-114F indicate that respective computer spaces 112D-112F do not have access to the computer object. The access-controlling user provides access to any of computer spaces by graphical activation of the corresponding one of access control indicators 114A-114F. It will be appreciated that access control indicators 114A-114F may be implemented in any of a wide variety of graphical control formats.

**[0022]** User interface also includes controls 116A and 116B with which the access controlling user can accept or cancel access control changes, and optionally may include a summary region 118 listing a summary of information from various fields in the user interface.

**[0023]** In one implementation, the listing of computer spaces 112 in access control field 110 is ordered as a ranked listing of possible destinations for the computer object. The ranked listing may be based upon the computer spaces that the access-controlling user has most recently accessed or interacted with. In another implementation, the ranked listing may be based on the extent or strengths of associations to the object, such as associations to computer locations commonly used by the user, or people the user commonly works or communicates with, or user computer interactions that are identified automatically. Such associations and their relative strengths could be determined automatically, as described below in greater detail.

**[0024]** Fig. 2 is an illustration of another computer object access control graphical user interface 130 for setting computer locations where a computer object may be accessed and users who may access the computer object. Access control user interface 130 includes a name field 132 in which a name for the computer object can be entered, either automatically by a computer or manually by the user. An author field 134 specifies who created the computer object (e.g., a computer file, a computer message, person contact information, etc.) for which access is being set with user interface 130.

**[0025]** An optional comment field 136 allows the access-controlling user to enter and store a comment about the object, and an optional "more" link 308 gives the access-controlling user optional additional fields for entering other properties relating to the object. An object type field 139 provides a type indication (e.g., graphical) for the computer object, specifying for example that the object is a word processor document, a picture, a spreadsheet, an audio or video file, etc.

**[0026]** An access control field 140 includes a listing of multiple computer spaces 142A, 142B (referred to generally as computer spaces 142) that have or can be selected to have access to the computer object. Computer spaces 142 are like computer spaces 112 (Fig. 1) and may

refer to different portions of a computer network (e.g., public and private portions or “folders”), different computers, different computer access rights of different computer users, different computer users, etc. Access control indicators 144A, 144B are analogous to access control indicators 114 and indicate whether each corresponding computer space 142A, 142B has been selected to have access to the computer object.

**[0027]** Access control graphical user interface 130 further includes a category field 146 that includes a flat listing of categories 148A-148C in which the computer object is or may be selected to be included according to category control indicators 149A-149C. In one implementation, categories 148A-148C of category field 146 correspond to directories or folders of a hierarchical file structure of the type commonly used in the art.

**[0028]** The hierarchical file structure, or tree structure, is commonly rendered as successively indented file structure levels for successively nested levels of folders or directories. In the flat representation of category field 146, folders and directories are listed by their names as categories in a single level (i.e., flat) arrangement. This flat arrangement allows multiple folder or directories to be listed efficiently without requiring the display area to render the hierarchical file structure. In another implementation, categories 148 may also or alternatively correspond to keywords for organizing objects. Categories 148 may be ranked or ordered according to the recency with which they have been used or accessed by the user.

**[0029]** Fig. 3 is a an illustration of a computer object access control graphical user interface 150 for accessing a computer object computer from a location on a computer or received via computer communication.

**[0030]** Access control user interface 150 includes a query field 152 in which a query can be entered to retrieve one or more computer objects. Query field 152 may be configured to receive a general database-type query (as illustrated) or may be a graphical user interface to a

hierarchical tree structure as is known in the art. As a database-type query, query field may receive or accommodate a chunk expression language format, for example.

**[0031]** As is known in the art, a chunk expression language format is a pseudo-natural language (e.g., English-like) that can be easily parsed by a computer into a formal query specification language, such as SQL. A sample chunk expression might be:

Show every email whose sender is 'John Smith' and whose subject contains "patent",

which would be parsed by a computer into a formal query specification language, such as SQL as:

SELECT \* FROM EMAIL WHERE SENDER = 'JOHN SMITH'  
AND SUBJECT LIKE '%patent%'

In general, a chunk expression might take the form of:

Show <selector> <object-type> whose <property>  
<comparator> <value> \* [<conjunction> <property>  
<comparator> <value>]

**[0032]** A recent object field 154 lists in reverse chronological order computer objects 156 that have been accessed by the user. Recent object field 154 includes both objects obtained from computer locations, such a computer files, pictures, audio files, etc., and computer communication objects such as emails or other computer messages, which the user can select to access at access indicators 158. A drop-down control 160 allows a user to access recent past queries.

**[0033]** Fig. 4 is a flow diagram of a computer object access control user interface method 180. User interface method 180 is performed in response to computer software for that is stored in a computer readable medium for performing the steps of method 180.

**[0034]** In step 182, a user initiates an access control method for a computer object.



- [0035]** In step 184, any computer spaces that have already been given access to the computer object are identified. In Fig. 1, for example, access control indicators 114A-114C may indicate that computer spaces 112A-112C had already been given access to the computer object.
- [0036]** In step 186, a ranked listing of plural other spaces that could be given access to the computer object is determined. The ranked listing may be based upon which computer spaces were most recently used or interacted with, or may be based upon associations determined between the computer object and the various computer spaces.
- [0037]** In step 188, a computer object access control graphical user interface is rendered on a computer display screen.
- [0038]** In step 190, access control entries from an access-controlling user are received into the user interface and accepted by the user.
- [0039]** In step 192, access to the computer object is provided to the computer spaces indicated by the user.
- [0040]** It will be appreciated that method 180 and user interface 100 could also be used to remove access from one or more computer spaces, such as by the user de-selecting the computer space from the access field of user interface 100. For some computer spaces, such as a computer space corresponding to an email address of another user, de-selecting of access rights might not be feasible because the object has already been transmitted. In this situation, de-selection of the computer space may be indicated as unavailable in the computer space.
- [0041]** Fig. 5 is a functional block diagram of an automatic context association system 200 that automatically determines context associations between user computer activities, computer objects (e.g., files, contacts, etc.), and other events. The context associations allow context association system 200 to provide users with automatic access to related computer files, contacts, and information (referred to generally as objects).

**[0042]** As described below, context associations between the objects may include similarities between them and their importance. The importance of and similarities between objects may be determined from user computer interactions with the objects. The user computer interactions may occur on a conventional desktop or laptop computer, or any other user-operated computing device such as a handheld computer, a digital cellular telephone, etc. For purposes of simplicity, these types of devices are referred to herein generally as computers. Context association system 200 may be implemented as software that is stored on and executed by one or more such computers.

**[0043]** Context association system 200 includes a data collection system 202 that collects data relating to user computer activities, computer actions, interactions between the user and various computer objects (e.g., applications, files, communications, etc.), and other events. For example, data collection system 202 may collect data relating to computer system events (e.g., the operation of applications or other software elements, various operations on computer files, sharing of computer data or other objects, etc.) computer communications (e.g., email communications, instant messaging communications, etc.), and external data (e.g., news, events, etc.)

**[0044]** In one implementation, data collection system 202 may include a system activity monitor 204, for example, that monitors user activity on a user-operated computer (not shown). System activity monitor 204 may be implemented as a computer program to which other applications, or the operating system, provide information of user computer activities, such as search queries, the opening or modification of a document or other file, etc. For example, while objects may exhibit many properties, searches performed by a user and the search results can indicate which properties are most important for the user. For each such activity, system activity monitor 204 creates or adds to a record of information

(sometimes called “metadata”) that is stored in a computer system database 206.

**[0045]** For example, system activity monitor 204 may include a universal input activity monitor that provides a single entry point for all user interactions with the computer, or at least specified user interactions with the computer. In this universal input implementation, system activity monitor 204 may determine and store as metadata in computer system database 206 any or all of the following: input method data (e.g., text, speech, mouse, etc.), input language data (English, Chinese, etc.), input mode data (e.g., dictation or typed command), input assistance data (e.g., typed word completion, command guessing, etc.), input feedback data (e.g., whether the computer needs additional information, corrections of wrong guesses by the computer, etc.), etc. For instance, if a user copies or stores a picture in a web browser, system activity monitor 204 could store the network address (e.g., Uniform Resource Locator, URL) of the source page, as well as the URL of the image itself, as metadata in computer system database 206.

**[0046]** Computer system database 206 stores information (e.g., “metadata”) about computer files stored on and activities carried out on a computer. The metadata may include conventional information, such as is conventional for computer-generated documents, including when the file was created, who created it, and a modification history. Also, some types of computer file types may have specific metadata fields that are not applicable to other file types. For instance, the metadata fields for a photo file type may include the size of the image, where the picture was taken, who took it, who is in it, etc. In one implementation, new metadata fields can be added to database system 206 to accommodate new file types or to store new information about an existing file type.

**[0047]** Table 1A lists exemplary information that may be obtained by system activity monitor 204 and stored in computer system database 206. With regard to the various computer files and computer information (referred

to collectively as objects) a computer user might interact with, system activity monitor 204 may collect data relating to any or all of the creator/author of the object, the type of the object, any pointer to the object (if available), activation events, object size, etc. For example, system activity monitor 204 may obtain the information about the objects with which a computer user interacts, determine whether the objects are new or unique relative to other objects listed in computer system database 206 (e.g., based upon a unique object identifier), and aggregate or store the activities in computer system database 206 in association with the unique object identifier of the object.

**Table 1A - Data Collection and Analysis**

Constructs	Operationalization	
	Data Collecting	Data Processing
<b>Objects Users Interact With</b>		
<b>General</b>	creator/author of object, type of object, pointer to object if it still exists, activation events, object size	get information about objects, determine if new/unique & aggregate events at level of unique object ID
<b>People</b>		
people in user's email	email client aliases	
people in user's contact list	contact list files	
people users IM, Chat with	buddy list email aliases	
<b>Communications</b>		
Emails	email events (open, close), email addresses	
phone messages	phone message events (open, close), email alias	
instant messages	instant message (IM) session events (open, close), people	
chat messages	chat session events	
<b>Files</b>		
any file created, opened, saved, played, sent, etc.	file names, place in directory	
<b>Web items</b>		
web pages	URLs	
streaming items played	URLs	
<b>Notifications</b>	type of notification, response notification	
<b>Applications</b>	pointer to application	

[0048] The objects with which the computer user interacts may be associated with other people, communications with other people, computer files, network or Web items, computer system notifications, and applications run on or accessible by the computer. System activity monitor 204 can obtain and store in computer system database 206 for each of the listed types of exemplary information.

[0049] Context association system 200 includes a relationship processing system 208 that determine one or more relationships between objects or

metadata obtained by data collection system 202 or stored in computer system database 206. In one exemplary implementation, relationship processing system 208 includes a similarity or association system 210 that determines a degree of similarity or association between at least a pair of computer files or objects. Similarity or the degree of association may be represented, for example, as a numeric value between 0 and 1, where 1 would be identity between objects and 0 would indicate they are completely unrelated. Association system 210 may be implemented as one or more programs that, for a given target file and a collection of other files, return a similarity ranking between them.

**[0050]** It will be appreciated that different types of documents, files, or objects can have different types of similarity. For instance, two photos can look similar, could be taken at similar times, could be of a similar person, place, or event, could be the same color, could have been sent to the same people or printed together, could be parts of the same photo (e.g., one could have been cut from the other), etc. Table 1B lists exemplary associations or similarities that association system 210 may identify and store in computer system database 206. As indicated, the associations or similarities may relate to objects or files having shared content, occurring at similar times or similar computer locations (e.g., computer, directory, etc.), being sent to or received from a common person, are be linked together.

**Table 1B - Data Collection and Analysis**

Constructs	Operationalization	
	Data Collecting	Data Processing
<b>Context: Associations between Objects</b>		
<b><i>Having shared content</i></b>		
	cut/paste event, insert event, objects	
	saved as event, objects	
<b><i>Occurring near in time</i></b>		
open on same machine at same time	open/close activation events, objects, date time of events	distance across times, overlapping sessions
<b><i>Occurring near in "place"</i></b>		
on same machine	machine name	
near each other in directory	place in directory structure	min number of jumps through tree from one to the other
near each other in web	link event, links from, links to	store metadata in computer system database
on the same email	attachment events, objects	aggregate at level of communication
<b><i>Being sent/received to one another</i></b>		
item sent to another person	sent event, objects	
pointer of item sent to another person	insert event, objects	
opened by a person	open event, objects	
saved by a person	save event, objects	
<b><i>Activating one from within the other</i></b>		
items downloaded from a web page	download event, objects	
linked via a URL	link event, objects, link from, link to	

In one implementation, association system 210 could be instructed to ignore certain types of similarity, or to weight one type of similarity more heavily than another. Moreover, the associations or similarities tracked and identified by association system 210 could be extensible.

[0051] In addition, objects with which a user interacts can be of a different importance or significance to the user. For instance, a document that a

user keeps open and edits of an extended period could be deemed to be of greater importance than a document that is opened for only a brief period of time. Likewise, a photo that is accessed and opened repeatedly could be deemed to be of greater importance than a photo that is only opened once. Table 1C lists exemplary factors by which association system 210, for example, could determine and store in computer system database 206 the relative importance of objects. As indicated, the importance factors may relate to the duration of user interactions, the frequency of interactions, the recency of interactions, as well as the immediacy of user responses, the amount of copying, duplication, and backups, and the extent objects are shared.

**Table 1C - Data Collection and Analysis**

Constructs	Operationalization	
	Data Collecting	Data Processing
<b>Importance of Objects</b>		
<b>Duration of interaction</b>		
	open and close activation events, object	session times calculated by min max time/date, then summed by object
	keyboard events, mouse events, object	summed by object
<b>Frequency of interaction</b>		
	activation events, date/times	count of unique activations/sessions
<b>Recency of interaction</b>		
	activation events, date/times	get maximum date
<b>Immediacy of response to</b>		
		difference in time
<b>Copying, duplication, backups</b>	print, save as, multiple versions	
<b>Sharing</b>	show during meeting, share via email, etc.	

[0052] Relationship processing system 208 may further include a chunking system 212 that uses the degrees of similarity or association determined by similarity or association system 210 for an arbitrarily large set of



objects or files and groups or "chunks" them into a specified number of sub-groups. For instance, given 200 photos, chunking system 212 could separate them into any number of sub-groups or chunks based on one or more criteria such as the dates they were taken, or who is in the photos, for example. The criteria to be used by chunking system 212 may be pre-defined or specified by a user, for example, and in one implementation can place the same item into several chunks (e.g., a picture of Bob and Sue would go into a Bob chunk as well as a Sue chunk). Chunking system 212 may be implemented as one or more programs stored on the computer.

**[0053]** A context association user interface system 216 utilizes information stored in computer system database 206 by system activity monitor 204, as well as information determined by similarity or association system 210 and chunking system 212 to display visualizations 218 illustrating context associations between computer system files, computer information, and other information. Visualizations 218 may be based upon the results of database queries 220 to which, in some instances, data filters 222 are applied. In addition, user interface system 216 can provide application and system notifications 124, as described below in greater detail.

**[0054]** Accordingly, context association system 200 accommodates innate human memory, which works by association so that events are often remembered as happening in relationship to other events. In contrast, conventional computer file systems force users to use and remember arbitrary, out-of-context filing structures. With the significant numbers of computer files and objects that many users interact with, conventional out-of-context filing structures can make it difficult, or impossible, for users to find and access computer information that was worked on or viewed in the past. Context association system 200 allows a user to view and assign searchable metadata and to retrieve documents, objects or files that match selected metadata.

- [0055]** Fig. 6 is a detailed block diagram of one implementation of data collection system 202. In addition to system activity monitor 204 for monitoring user computer activity, for example, data collection system 202 can also include any or all of the following, including an event handler 232 through which one or more applications 234 (including other software services) send metadata information about application events to data collection system 202 and receive metadata information about application events from data collection system 202. Applications 234 may be separate from an underlying operating system for the computer, or may be services or features provided by the operating system.
- [0056]** A data collection container 236 operates to control system activity monitor 204, event handler 232, and any other data collection modules, and particularly controls data collection relating to system activity, copy and paste between documents, incoming and outgoing messages, opening shared files, etc. In addition, data collection container 236 can operate to obtain or receive common data (like weather, news), other than data relating specifically to operation or use of the computer, from one or more separate computer information sources 238 (e.g., public servers).
- [0057]** Applications 234 running on a computer with automatic context association system 200 could register with event handler 232 to receive enhanced document management functionality. In particular, data collection container 236 supports cross-application integration that allows collection of data or information about cross-application events, such as copy and paste between documents, attaching a document to an email message, etc., thereby allowing relationships between objects to be determined or derived by relationship processing system 208. Moreover, applications 234 can interact directly with data collection system 202 to provide contextual information not readily available to system activity monitor 204. For example, if an email attachment is saved to disk as a separate file, the email application 234 may create a link between this file

and the original email message (and, by extension, with the message sender). Control system activity monitor 204 and event handler 232 allow data to be collected at high efficiency and at a high granularity level, and with the minimal amount of processing required. In addition, data collection system 202 in one implementation may be extensible to allow for the installation of additional data collectors that may be developed in the future.

**[0058]** In one implementation users are notified that the data is collected and what it is used for. Users are provided one or more controls to selectively enable or disable the data collection, or explicitly adjust the importance of objects and their association weights, as described below. In other implementations, the ways in which the data is collected may be modified automatically based on how the data is used. While data is typically collected at the level of an individual user and computer, it will be appreciated that the data from multiple users and their computers could be combined. For such a shared data implementation, system 200 will provide unique identifiers for the data, objects and sessions, consistent handling of to- and from- fields for communications, replication of event identifiers across computers in shared sessions, privacy, and so forth.

**[0059]** In accordance with the practices of persons skilled in the art of computer programming, the present invention is described above with reference to acts and symbolic representations of operations that are performed by various computer systems and devices. Such acts and operations are sometimes referred to as being computer-executed and may be associated with the operating system or the application program as appropriate. It will be appreciated that the acts and symbolically represented operations include the manipulation by a CPU of electrical signals representing data bits, which causes a resulting transformation or reduction of the electrical signal representation, and the maintenance of data bits at memory locations in a memory system to thereby reconfigure

or otherwise alter the computer system operation, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, or optical properties corresponding to the data bits.

**[0060]** Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be modified in arrangement and detail without departing from such principles. In view of the many possible embodiments to which the principles of our invention may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather, I claim as my invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.